

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-139049

(43)公開日 平成6年(1994)5月20日

(51)Int.Cl.⁵

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 7/38

Y 9291-5B

15/18

8945-5L

G 0 6 G 7/60

審査請求 未請求 請求項の数3(全 7 頁)

(21)出願番号

特願平4-292443

(22)出願日

平成4年(1992)10月30日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(71)出願人 000233169

株式会社日立マイコンシステム

東京都小平市上水本町5丁目22番1号

(72)発明者 岡橋 卓夫

東京都小平市上水本町5丁目22番1号 株

式会社日立マイコンシステム内

(72)発明者 浅井 光男

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

(74)代理人 弁理士 小川 勝男

最終頁に続く

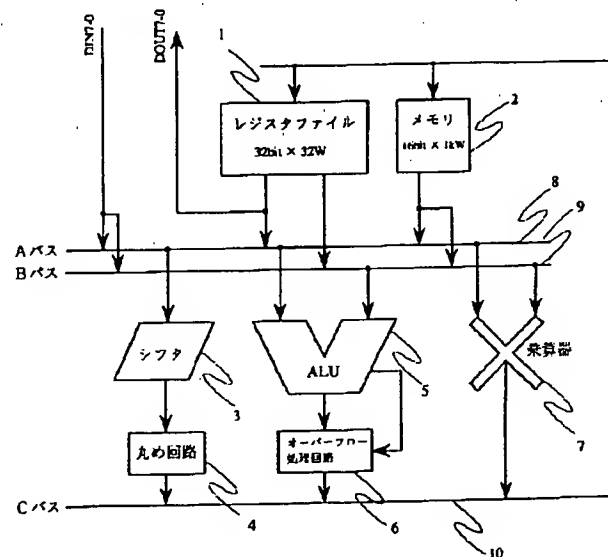
(54)【発明の名称】 並列データ処理装置

(57)【要約】

【構成】 演算器の出力にオーバーフロー処理回路6を設ける。オーバーフロー処理回路6で正のオーバーフローが発生した時、表現できる正の最大値、負のオーバーフローが発生した時表現できる負の最大値、その他の時は演算器の出力を演算結果とする。演算の精度を高める方法としてシフトの出力に丸め回路を設け算術右シフトを実行した時に零方向へ丸める。

【効果】 オーバーフローが発生したときに最大値が演算結果として返る為此の値を用いて演算を続行することができる。零方向へ丸める事により繰り返し演算で正しい値に収束するようになる。

図1



【特許請求の範囲】

【請求項1】複数の演算部と制御部より構成される並列データ処理装置に於て、演算部の演算結果が正のオーバーフローを発生した時、表現出来る正の最大値を演算結果とし演算を続行する並列データ処理装置。

【請求項2】請求項1記載の並列データ処理装置に於て、演算部の演算結果が負のオーバーフローを発生した時、表現出来る負の最大値を演算結果とし演算を続行する並列データ処理装置。

【請求項3】請求項1記載前記並列データ処理装置に於て、演算部にシフトを有し、シフトにて算術右シフト実行時、零方向への丸めを行なうことを特徴とする並列データ処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、並列データ処理装置に係り、特にニューラルネットワークの演算結果の処理方法に適用して有効な技術に関するものである。

【0002】

【従来の技術】従来のデータ処理装置では、四則演算等で演算結果が表現出来る最大値を越えた（オーバーフロー）場合、これを検出してオーバーフロー処理を行なう。その時にオーバーフローエラー発生で、止めてしまう場合や、無限大という非数を演算結果とする場合などがある。

【0003】また、データの算術右シフトを実行した場合、溢れたビットは切り捨て（ $-\infty$ 方向への丸め）していた。

【0004】

【発明が解決しようとする課題】多数の演算器が同時に動作する並列演算処理装置に於ては、この中の1つの演算器でもオーバーフローを発生しエラーで停止すると全ての演算器が停止してしまう問題点がある。また、無限大という非数を演算結果とした場合演算の続行ができず停止してしまう問題点がある。

【0005】また、繰り返し演算を実行して収束させる場合、演算が（負数） \times （1に近い1未満の数）の結果を $-\infty$ 方向へ丸めると本来収束すべき値と異なる値に収束することが有るという問題点がある。

【0006】

【課題を解決するための手段】並列演算処理装置に次の機能を持たせる。

【0007】（1）演算結果が正のオーバーフローした時、演算器の表現出来る正の最大値を演算結果とし演算を続行する。

【0008】（2）演算結果が負のオーバーフローした時、演算器の表現出来る負の最大値を演算結果とし演算を続行する。

【0009】（3）算術右シフト実行時、零方向への丸めを行なう。

【0010】

【作用】上述した手段（1）、（2）によれば、並列演算処理装置の演算器の結果が少なくとも1個以上オーバーフローするデータを処理した時オーバーフローを発生せずに表現出来る最大値を返してくる。この演算器はこの結果を用いて演算を進めていく。従ってオーバーフローの為の処理をすること無しに装置全体が演算を進めて行くことが出来る。

【0011】上述した手段（3）によれば演算が（負数） \times （1に近い1未満の数）の場合でも実行毎に変化してゆき本来の収束値に収束する。

【0012】

【実施例】図1に本発明の1実施例の並列演算処理装置であるニューロコンピュータにもちいる演算素子のブロック図を示す。本演算素子は、ニューロコンピュータのニューロンとして用いられる。本演算素子は、各種データや演算結果を格納するレジスタファイル1、ニューロの重み値を格納する為のメモリ2、シフト3、ALU5、乗算器7より構成される。

【0013】レジスタファイル1の大きさは、32ビット \times 32ワードである。入力は、Cバス10に接続されている。この出力は素子外部への出力DOUに接続されている。他方の出力はBバス9に接続されている。

【0014】メモリ2の大きさは、16ビット \times 1Kワードである。入力は、Cバス10に接続されている。出力は、Aバス8、Bバス9に接続されている。

【0015】シフト3は、算術/論理の右/左シフトを行なう。入力/出力は、32ビットである。入力Aバス8に接続されている。出力は丸め回路4を通してCバス10に接続されている。

【0016】丸め回路4はシフト3にて算術右シフト実行時に零方向の丸めを行なう。他の場合はビットを切り捨てる。

【0017】ALU5は、加算、減算、論理和、論理積、排他的論理和、を行なう。32ビット入力32ビット出力である。一方の入力は、Aバス8に接続されている。他方の入力、Bバス9に接続されている。出力は、オーバーフロー処理回路6を通してCバス10に接続されている。

【0018】オーバーフロー処理回路6は、正のオーバーフロー発生時に出力を7FFFFFFF HEX. とし、負のオーバーフロー発生時に出力を80000000 HEX. とする。

【0019】乗算器7は、16ビット \times 8ビット=24ビットの演算を行なう。16ビット入力Aバス8に接続されており8ビット入力Bバス9に接続されている。出力は、Cバス10に接続されている。

【0020】また、これらの構成要素は完全水平なコマンドにより動作する。各構成要素は、1マシンサイクルで動作する。従ってあるクロックのタイミングでレジス

タファイル等から読み出されたデータは演算実行されて次のクロックでレジスタファイル等に書込まれる。

【0021】図2にオーバーフロー処理回路の実施例を示す。選択回路は、32ビット3入力、32ビット1出力である。一番目の入力はALU出力、2番目の入力は80000000HEX、固定、3番目の入力は7FFFFFFFHEX、固定である。選択制御信号は、ALUの出力及びALUのステータス出力信号により作られる。正のオーバーフロー時には、7FFFFFFFHEX、負のオーバーフロー時には、80000000HEX、その他の時にはALUの出力が選択される。

【0022】図3に本発明の一実施例である並列演算装置であるニューロコンピュータのブロック図を示す。本装置は前述した演算素子11が16個と制御部12より構成される。制御部12は演算素子への命令を命令バス13へ出力し、演算素子へのデータをDINバス14へ出力する。制御部12は演算素子からのデータをDOU
Tバス15より入力する。制御部12は素子の選択信号やリセット信号等を制御バス16へ出力する。演算素子11は命令を命令バス13より取り込む。演算素子11はデータをDINバス14より取り込む。演算素子11はデータをDOU
Tバス15へ出力する。演算素子11は各種制御信号を制御バス16より受け取る。

【0023】以下、図1と図3を用いてオーバーフロー発生時の動作を説明する。最初は、制御バスの素子選択信号を用いて各素子のレジスタ、又はメモリに初期値を入力する。この時、選ばれた素子はDINバスよりデータを入力しALU5を通してレジスタ又はメモリにデータを格納する。但しこの時ALU5は演算を行なわ
ない。

【0024】全ての素子に初期値が設定された後、全ての素子に同一の命令が命令バスを通して与えられる。演算の実行が進みある素子の演算器の入力が次の値となった場合を例として説明する。ALU5の一方の入力が4FFFFFFFHEX、で他方の入力が50000000HEX、で演算内容が加算である場合を考える。この時の演算結果は、9FFFFFFFHEX、と成り32ビットで表現できない値と成り正のオーバーフローである。しかし、本演算器では、正のオーバーフローの場合7FFFFFFFHEX、を結果として返す。この後の演算は7FFFFFFFHEX、を用いて実行する。こ
こで誤差が生じるが繰り返し演算である値に収束させる場合には、徐々に補正が行なわれ最終的には正しい値に収束する。

【0025】負のオーバーフローが発生したときも同様にして演算結果として80000000HEX、を返し、これを用いて演算を進めて行く。

【0026】この様に本素子を用いた並列演算装置では、オーバーフローが発生しても実行を続行することができる。

【0027】図1を用いて丸めの動作を説明する。固定小数点演算で乗算を繰り返す場合について説明を行なう。小数点位置は下位より4ビット目と5ビット目の間にある。乗算器7の16ビット入力がFFF8HEX、(-0.5)であり8ビット入力が0FHEX、(0.9375)の場合を考える。乗算器7の出力はFFFF87HEX、である。ここで桁合わせの為にシフト3に入力をする。この時入力値は、32ビットに拡張してFFFFFFFF87HEX、である。今、小数点は下位より8ビット目と9ビット目の間にある。従ってシフトで算術右4ビットシフトを行なえば桁を合わせた事に成る。ここで-∞方向への丸めを行なうと結果はFFFFFFFF8HEX、と成る。この値は16ビット入力値と同じである。即ち演算前の値と演算後の値が変化していない。従って、この値を再度16ビット入力として用いても結果は変わらない。

【0028】本発明ではシフトした値を零方向へ丸める。この場合の結果はFFFFFFFF9HEX、となり入力値から変化している。この値を再度16ビット入力として用いて繰り返し演算を行なっていくと最終的な答えは零になる。

【0029】この演算を無限精度で実行したとき結果が零になることは明らかである。-∞方向への丸めでは最初の入力値と変わらないが零方向への丸めでは結果が零となり真の値と等しい事がわかる。

【0030】以上、特定のデータについて説明を行なったが他のデータについても同じである。

【0031】

【発明の効果】本発明において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば以下のとおりである。

【0032】即ち、オーバーフローになるデータを容易に取り扱える並列データ処理装置を構築することができる。

【0033】また、演算精度の高いデータ処理装置を構築することができる。

【図面の簡単な説明】

【図1】本発明の一実施例である並列データ処理装置を構成する演算素子のブロック図である。

【図2】図1で示すオーバーフロー処理回路の詳細図である。

【図3】本発明の一実施例である並列データ処理装置のブロック図である。

【符号の説明】

1…レジスタ、2…メモリ、3…シフト、4…丸め回路、5…ALU、6…オーバーフロー処理回路、7…乗算器、8…内部バスA、9…内部バスB、10…内部バスC、11…演算素子、12…制御ブロック、13…命令バス、14…データ入力バス、15…データ出力バス、16…制御信号。

【図1】

図1

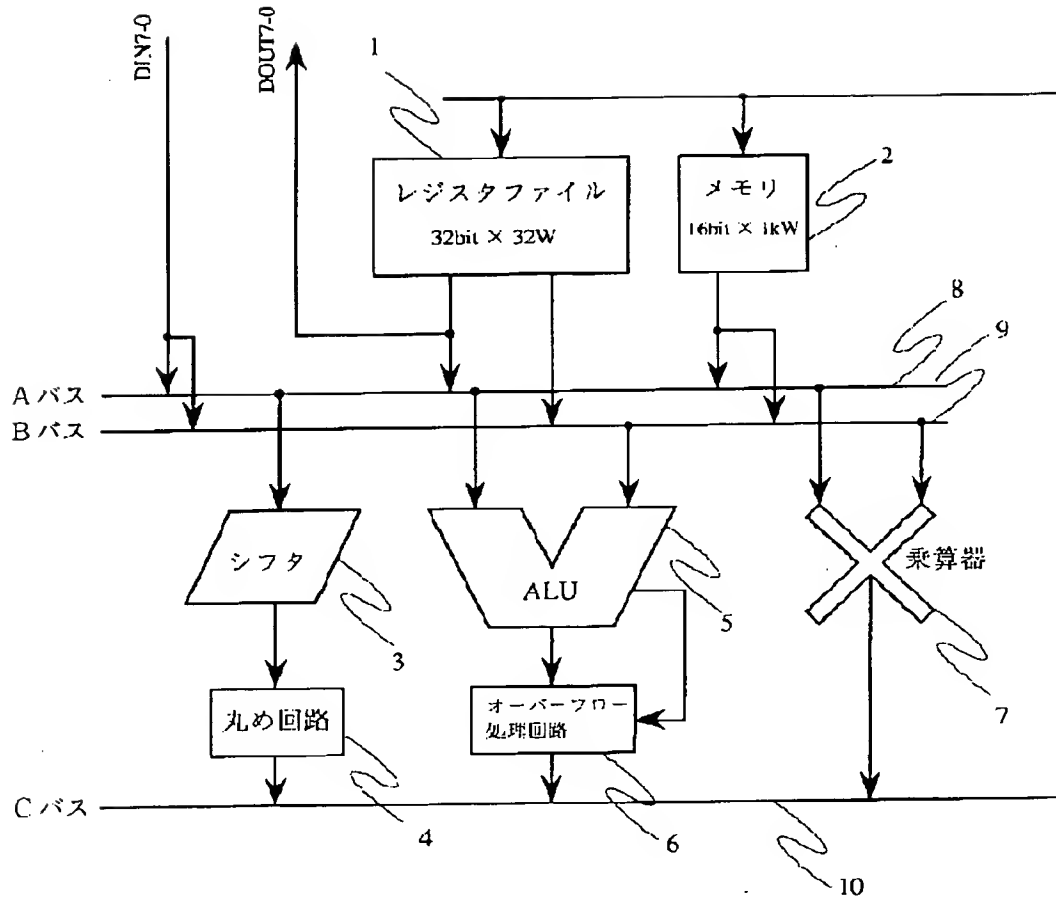
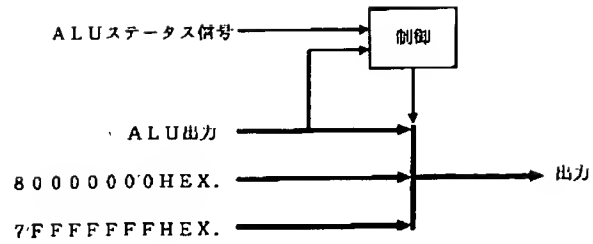


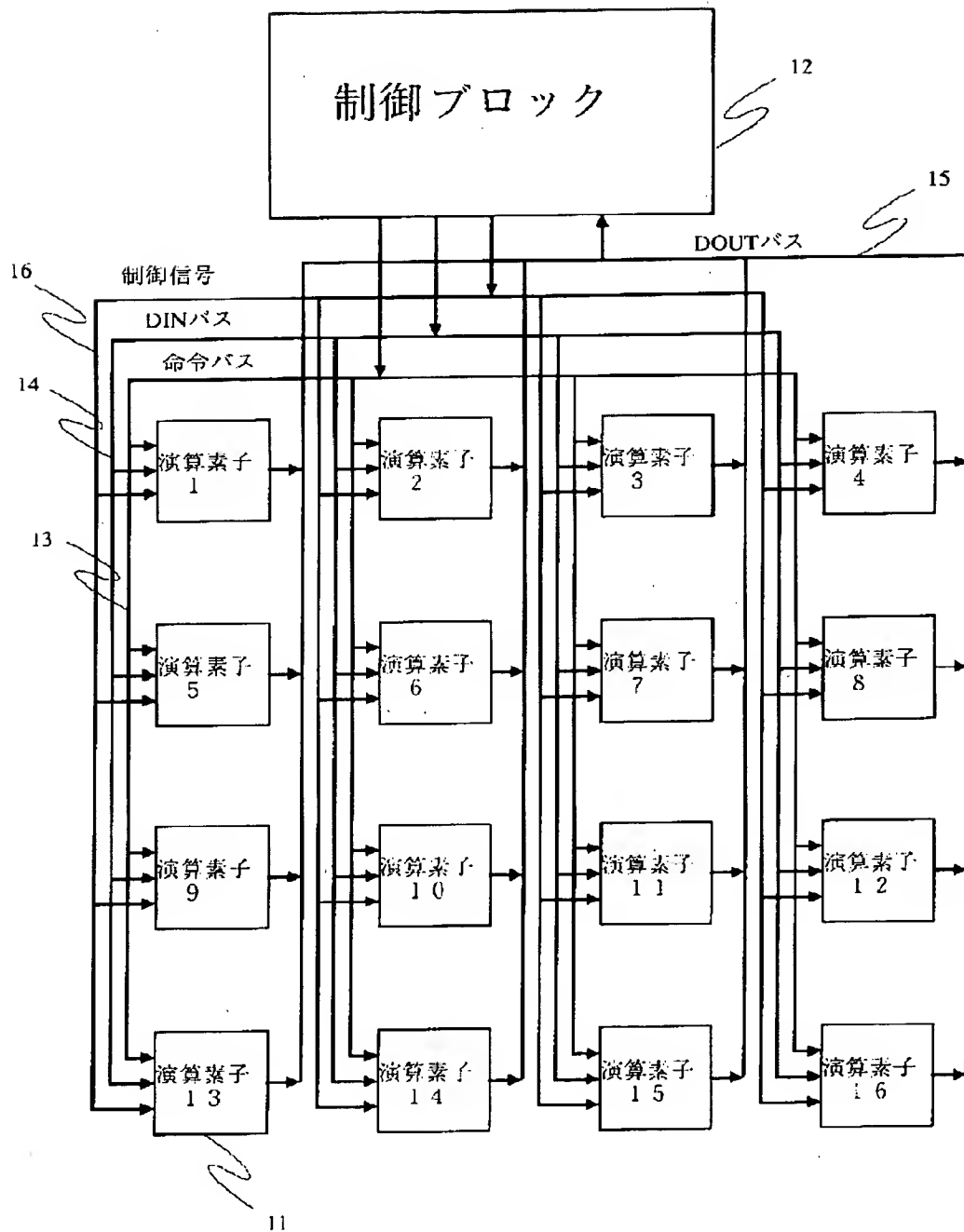
图 2



制御信号	出力
正のオーバーフロー	7 F F F F F F F H E X.
負のオーバーフロー	8 0 0 0 0 0 0 0 H E X.
その他	A L U出力

【図3】

図3



フロントページの続き

(72)発明者 柴田 克成
 東京都国分寺市東恋ヶ窪1丁目280番地
 株式会社日立製作所中央研究所内

(72)発明者 佐藤 裕二
 東京都国分寺市東恋ヶ窪1丁目280番地
 株式会社日立製作所中央研究所内

(72) 発明者 橋本 雅
東京都小平市上水本町5丁目20番1号 日
立超エル・エス・アイ・エンジニアリング
株式会社内

(72) 発明者 坂口 隆宏
東京都小平市上水本町5丁目20番1号 日
立超エル・エス・アイ・エンジニアリング
株式会社内

(72) 発明者 ▲高▼柳 博
東京都小平市上水本町5丁目22番1号 株
式会社日立マイコンシステム内

(72) 発明者 茂木 啓次
東京都小平市上水本町5丁目22番1号 株
式会社日立マイコンシステム内

(72) 発明者 落合 辰男
東京都小平市上水本町5丁目22番1号 株
式会社日立マイコンシステム内

(72) 発明者 桑原 良博
東京都小平市上水本町5丁目22番1号 株
式会社日立マイコンシステム内